TK_59: Analysis of piRNA-encoding 3'UTRs

Hernan Lorenzi

10/04/2023

PI: Astrid Haase
Point of Contact: Parthena Konstantinidou
Contact email: parthena.konstantinidou@nih.gov

Summary: From Thenia's email *"To sum up what we discussed yesterday, we hope to systematically check whether the genes that we find upstream of our clusters are really expressed and have a possible alternative poly A signal which creates the longer versions preferably in the embryonic stage (E16.5) and not in later stages (P14). It would be also interesting to check if these genes have something in common."*

R code

```
library(GenomicRanges)
library(cowplot)
library(tidyverse)
library(DGEobj.utils)
library(GenomicFeatures)
library(scales)
```

Load libraries

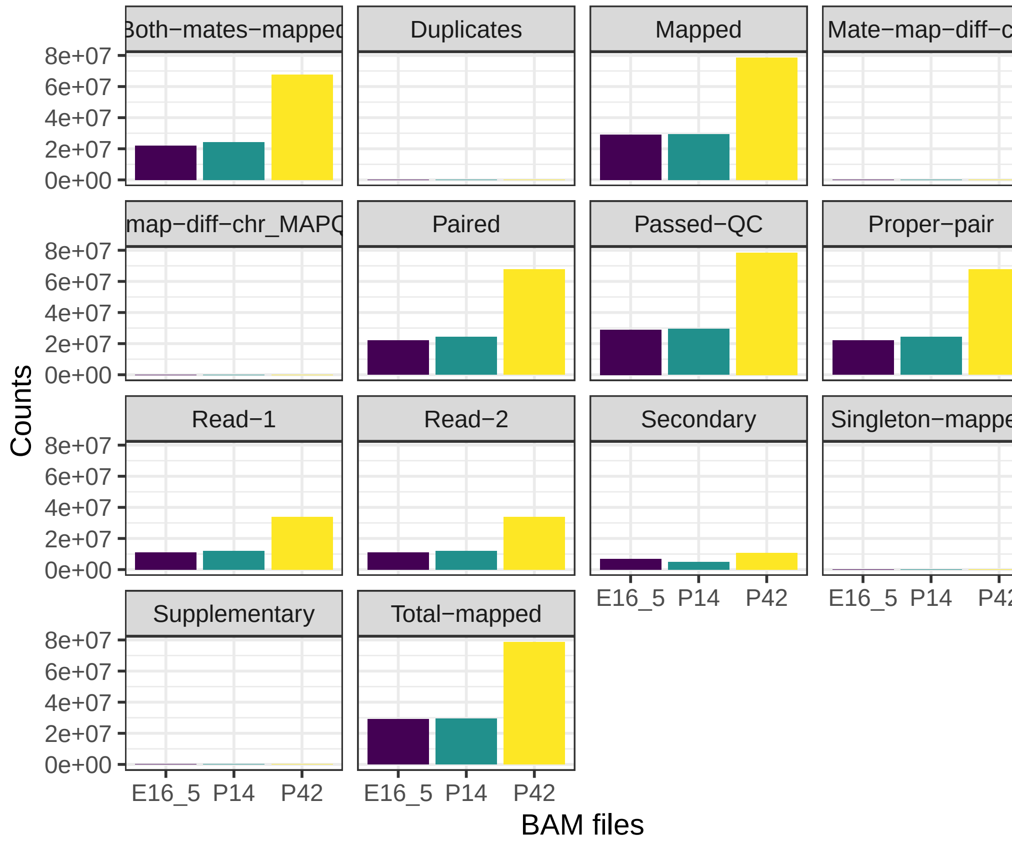
```
if( file.exists("./data/tk_59_environment.Rdata") ){
  load(file = "./data/tk_59_environment.Rdata")
}
```

Load environment, if exists

```
# This function outputs the total length on non-overlaping exons for each gene
# See https://www.biostars.org/p/83901/
get_transcript_sizes_from_gtf <- function(annotation_file, format = 'gtf'){
  if (format == 'custom'){

    # Calculate cluster lengths
    annotation_file$length_bp <- abs(annotation_file$Start - annotation_file$End)
    annotation_file <- tibble(gene_id = annotation_file$GeneID,
                              length_bp = annotation_file$length_bp)
    # Sum all lengths from clusters associated with the same gene
    exonic.gene.sizes.tb <- annotation_file %>%
                            group_by(gene_id) %>%
                            summarise(length_bp=sum(length_bp))
    # Comvert exonic.gene.sizes.tb to a list
    exonic.gene.sizes <- exonic.gene.sizes.tb$length_bp
    names(exonic.gene.sizes) <- exonic.gene.sizes.tb$gene_id
    return(exonic.gene.sizes)
  }else{
    # First, import the GTF-file that you have also used as input for htseq-count
    txdb <- makeTxDbFromGFF(annotation_file, format = format)
    # then collect the exons per gene id
    exons.list.per.gene <- exonsBy(txdb,by="gene")
    # then for each gene, reduce all the exons to a set of non overlapping exons, calculate their lengt
    exonic.gene.sizes <- sum(width(GenomicRanges::reduce(exons.list.per.gene)))
  }
  return(exonic.gene.sizes)
}

# Function to calculate tpm and fpkms
normalize_by_TPM <- function(read_counts_column, annotation_file, annot_file_format = 'gtf') {

  transcripts_length <- get_transcript_sizes_from_gtf(annotation_file = annotation_file,
                                                      format = annot_file_format)

  c.tb <- tibble(gene_id = rownames(read_counts_column),
                 read_counts = as.vector(read_counts_column[,1]))

  # Eliminate gene IDs from counts.df without transcript length info in transcript_lengths
  tl.tb <- tibble(gene_id=names(transcripts_length), length_bp=transcripts_length)
  tl.tb <- filter(tl.tb, !is.na(length_bp))

  # Merge read counts and transcript length tibbles
  merged_tibble <- inner_join(x = c.tb,
                              y = tl.tb,
                              by = join_by(gene_id))


  # Calculate TPMs
  merged_tibble <- merged_tibble %>%
                   mutate(reads_per_kb = read_counts * 1000/ length_bp) %>%
                   mutate(tpm = reads_per_kb * 1e6 / sum(reads_per_kb))

  # Calculate FPKMs
  merged_tibble <- merged_tibble %>%
                   mutate(fpkm = read_counts * 1000 * 1e6 / length_bp / sum(read_counts))

  # See reference for formula:
  # https://www.reneshbedre.com/blog/expression_units.html
  # https://www.biostars.org/p/273537

  return(merged_tibble)
}
```

Useful functions

```
library(parseR) # For running samtools flagstat
if(! exists("E16_5.bam.flagstat")){
  E16_5.bam.flagstat <- run_samflagstat(samtools="/Users/lorenziha/miniconda3/envs/ARTDeco/bin/samtools
                                        bamfile = "./NEW_ARTDECO_ANALYSIS/ARTDeco_input/E16_5.bam")
}
if(! exists("P14.bam.flagstat")){
  P14.bam.flagstat <- run_samflagstat(samtools="/Users/lorenziha/miniconda3/envs/ARTDeco/bin/samtools",
                                        bamfile = "./NEW_ARTDECO_ANALYSIS/ARTDeco_input/P14.bam")
}
if(! exists("P42.bam.flagstat")){
  P42.bam.flagstat <- run_samflagstat(samtools="/Users/lorenziha/miniconda3/envs/ARTDeco/bin/samtools",
                                        bamfile = "./NEW_ARTDECO_ANALYSIS/ARTDeco_input/P42.bam")
}

# Group flagstat results
flagstat <- rbind(t(P42.bam.flagstat), t(P14.bam.flagstat), t(E16_5.bam.flagstat))
rownames(flagstat) <- str_remove(string = rownames(flagstat), pattern = "./NEW_ARTDECO_ANALYSIS/ARTDeco
colnames(flagstat) <- c("Total-mapped","Passed-QC","Secondary","Supplementary","Duplicates","Mapped","Pa

flagstat.tbl <- tibble("Total-mapped"=flagstat[,"Total-mapped"],"Passed-QC"=flagstat[,"Passed-QC"],"Sec

flagstat.tbl<- gather(data = flagstat.tbl, key = bam_file) %>% mutate( bam = rep(c("P42","P14","E16_5")

# Generate barchar plots
flagstat.p <- flagstat.tbl %>% ggplot(aes(bam, value, fill=bam)) + geom_bar(stat="identity") + facet_wra

ggsave2(filename = "./Plots/bamstats_1.pdf", plot = flagstat.p)
flagstat.p
```



Run samtools' stats on bam files

```
# NOTE: Granges works with subset() to filter rows by metadata values

#Granges of the piRNA clusters that could come from upstream genes (and the corresponding genes' names f

load(file = "./data/Mili_prepach_gene3end_regions.RData")

# Mm10_refGene_all_biotypes_curated_TK : Granges of the Mm10 refGene mouse annotation including both pr
load(file = "./data/Mm10_refGene_all_biotypes_curated_TK.RData")
```

Load Granges files

Make bed files for Mm10_refGene_all_biotypes_curated_TK The idea is to merge contiguous features to form exons, excluding intron coords.

```
# just make a smaller name removing intron features
coding_curated_ann <-subset(Mm10_refGene_all_biotypes_curated_TK, type != 'INTRON')
coding_curated_bed <- tibble(Chr=as.vector(seqnames(coding_curated_ann)),
                      Start=start(coding_curated_ann)-1, # to be bed compliant
                      End=end(coding_curated_ann),
                      Type=as.vector(coding_curated_ann$type),
                      GeneID=as.vector(coding_curated_ann$gene_name),
                      Strand=as.vector(strand(coding_curated_ann))
                      )
write.table(coding_curated_bed, file = "coding_curated.bed", sep = "\t", col.names = F, quote = F, row.n
write.table(unlist(Mili_prepach_gene3end_regions$Gene_ovrlp), file = "genes_of_interest.txt", sep = "\t
```

Write list of genes of interest

```
clust_bed <- as.data.frame(paste(seqnames(Mili_prepach_gene3end_regions),
            start(Mili_prepach_gene3end_regions),
            end(Mili_prepach_gene3end_regions),
            Mili_prepach_gene3end_regions$uniq_reads_FPM,
            round(Mili_prepach_gene3end_regions$fraction_of_width_covered_by_unique_alignments, digits
            strand(Mili_prepach_gene3end_regions)
            )
          )

colnames(clust_bed) <- c('data')
write.table(clust_bed, file = "clusters.bed", sep = "\t", col.names = F, quote = F, row.names = F)
```

Generate bed file for clusters

```
library(Rsubread)
```

Count reads per gene for Ensemble and RefSeq annotations

```
genes_of_interest <- unlist(Mili_prepach_gene3end_regions$Gene_ovrlp)
#subset(P14.refseq.counts$counts, rownames(P14.refseq.counts$counts) %in% genes_of_interest)
```

Count reads per feature

Normalize read counts

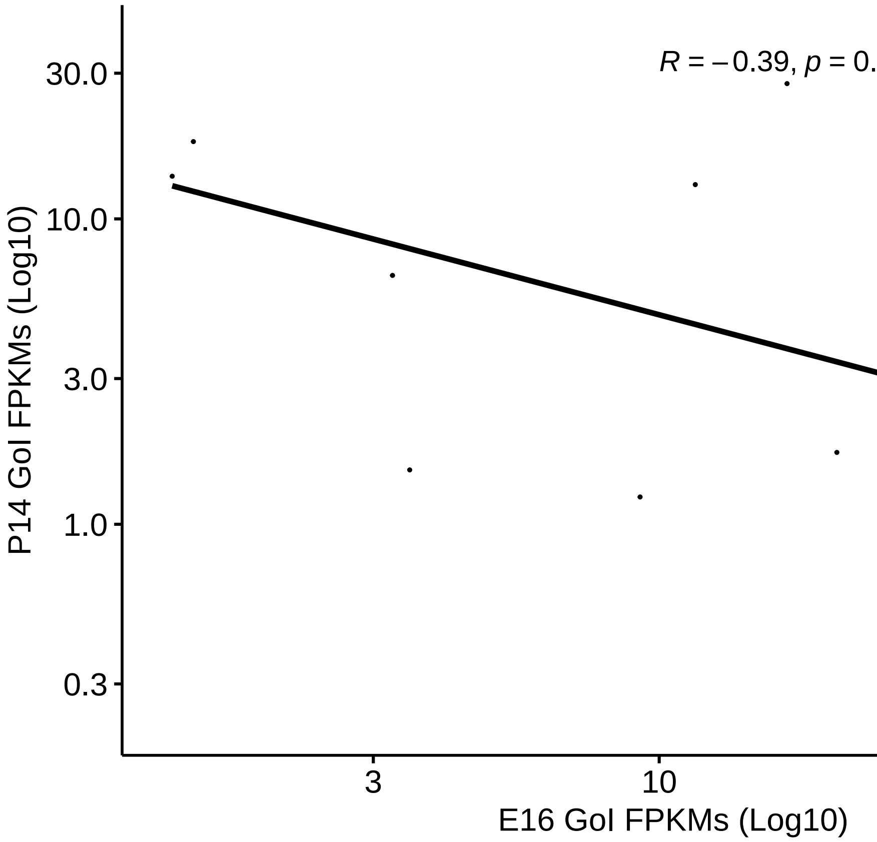
```
P14.counts <- normalize_by_TPM(read_counts_column = P14.refseq.counts$counts,
                               annotation_file = "./mm10.ncbiRefSeq.transcripts.gtf",
                               annot_file_format = 'gtf')

P42.counts <- normalize_by_TPM(read_counts_column = P42.refseq.counts$counts,
                               annotation_file = "./mm10.ncbiRefSeq.transcripts.gtf",
                               annot_file_format = 'gtf')

E16.counts <- normalize_by_TPM(read_counts_column = E16.refseq.counts$counts,
                               annotation_file = "./mm10.ncbiRefSeq.transcripts.gtf",
                               annot_file_format = 'gtf')
```

```
top10_GoI <- c("Zim2","D10Wsu102e","Gan","Elk4","Eif4ebp2","Dyrk1b","Zbtb37","Myl10","Frmd8","E130317F2

# Keep top-10 genes of interest
P14.counts.GoI <- filter(P14.counts, gene_id %in% top10_GoI) %>% mutate (group = "P14.GoI")
P14.counts.not_GoI <- filter(P14.counts, ! gene_id %in% genes_of_interest) %>% mutate (group = "P14.not

P42.counts.GoI <- filter(P42.counts, gene_id %in% top10_GoI) %>% mutate (group = "P42.GoI")
```

```
P42.counts.not_GoI <- filter(P42.counts, ! gene_id %in% genes_of_interest) %>% mutate (group = "P42.not_

E16.counts.GoI <- filter(E16.counts, gene_id %in% top10_GoI) %>% mutate (group = "E16.GoI")
E16.counts.not_GoI <- filter(E16.counts, ! gene_id %in% genes_of_interest) %>% mutate (group = "E16._no

# Append all tibbles
combined_counts <- rbind(P14.counts.GoI,P14.counts.not_GoI,P42.counts.GoI,P42.counts.not_GoI,E16.counts
```

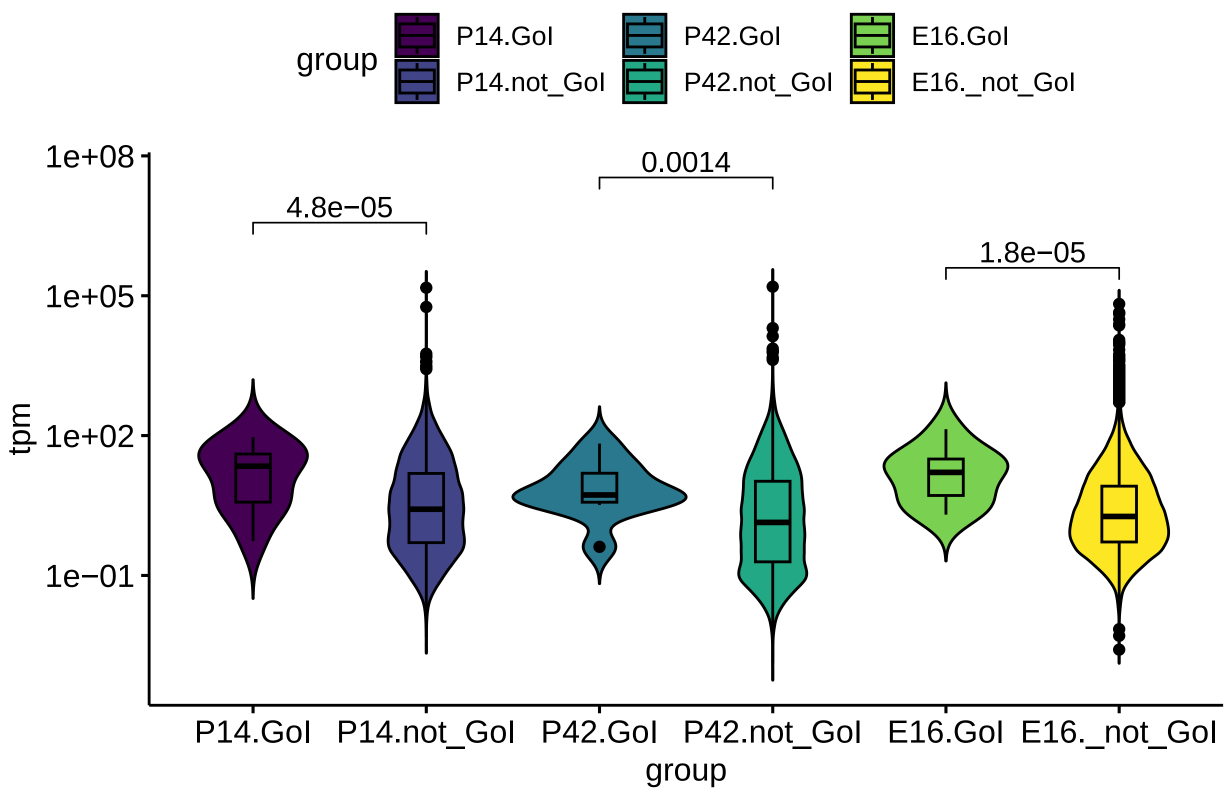
Subset genes of interest

```
library(ggplot2)
library(ggpubr)

dir.create(path = "./Plots", showWarnings = F)

join_counts <- inner_join(x = E16.counts.GoI, y = P14.counts.GoI, by = "gene_id")

p1 <- join_counts %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "E16 GoI FPKMs (Log10)",
                                ylab = "P14 GoI FPKMs (Log10)",
                                size = 0.5,
                                shape = 20, palette = "viridis",
                                add = "reg.line"
                                ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

ggsave2(filename = "E16Int_vs_P14Int_FPKM_corr.pdf", plot = p1, path = "./Plots")
p1
```



Plot correlation GoI between P14 and E16

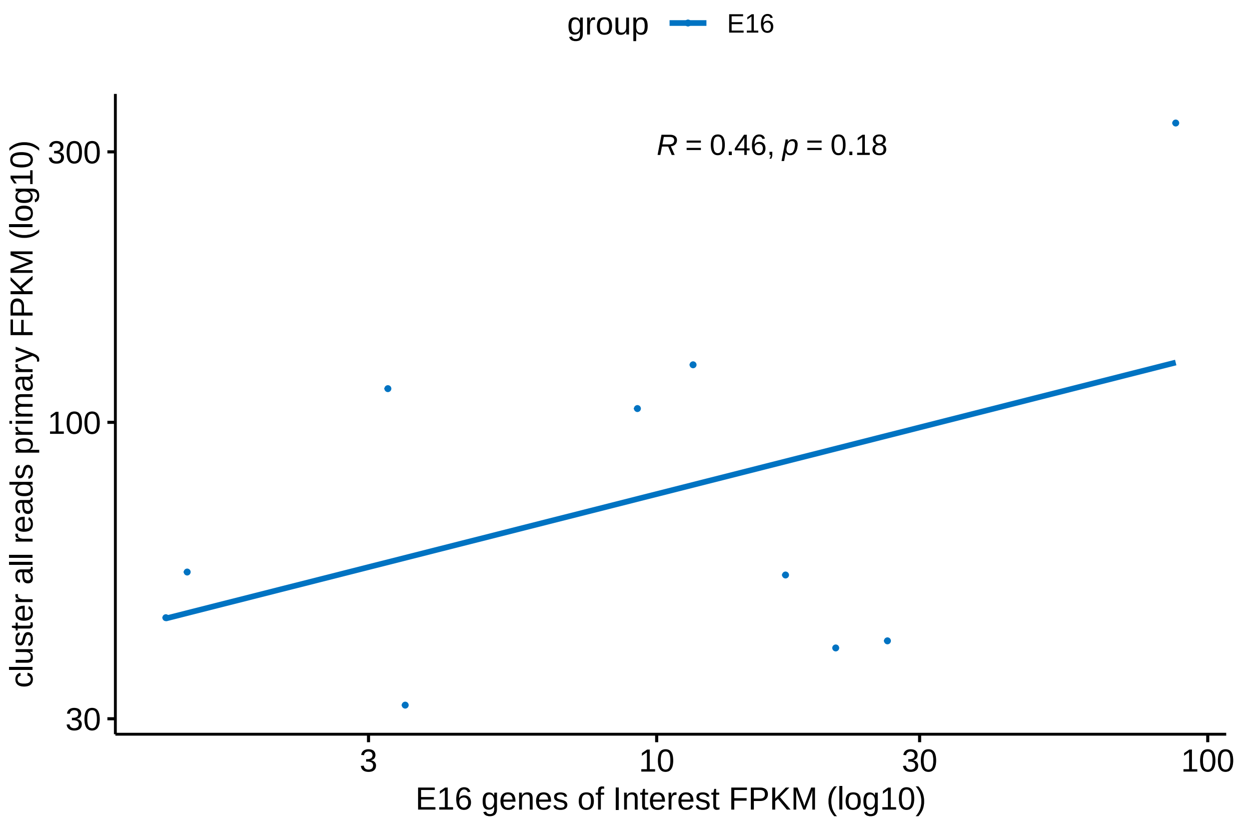
```
my_comparisons <- list( c("E16.GoI","E16._not_GoI"), c("P14.GoI", "P14.not_GoI"), c("P42.GoI", "P42.not_

p2 <- combined_counts %>% ggviolin(x="group",
                                   y="tpm",
                                   fill="group",
                                   draw_quantiles = T,
                                   add = "boxplot") +
                                   stat_compare_means(comparisons = my_comparisons) +
                                   scale_y_log10() + scale_fill_viridis_d(option = "D") +
                         labs(caption = "GoI = Top 10 genes")

ggsave2(filename = "Int_vs_nonInt_genes_per_group.pdf", plot = p2, path = "./Plots")
p2
```



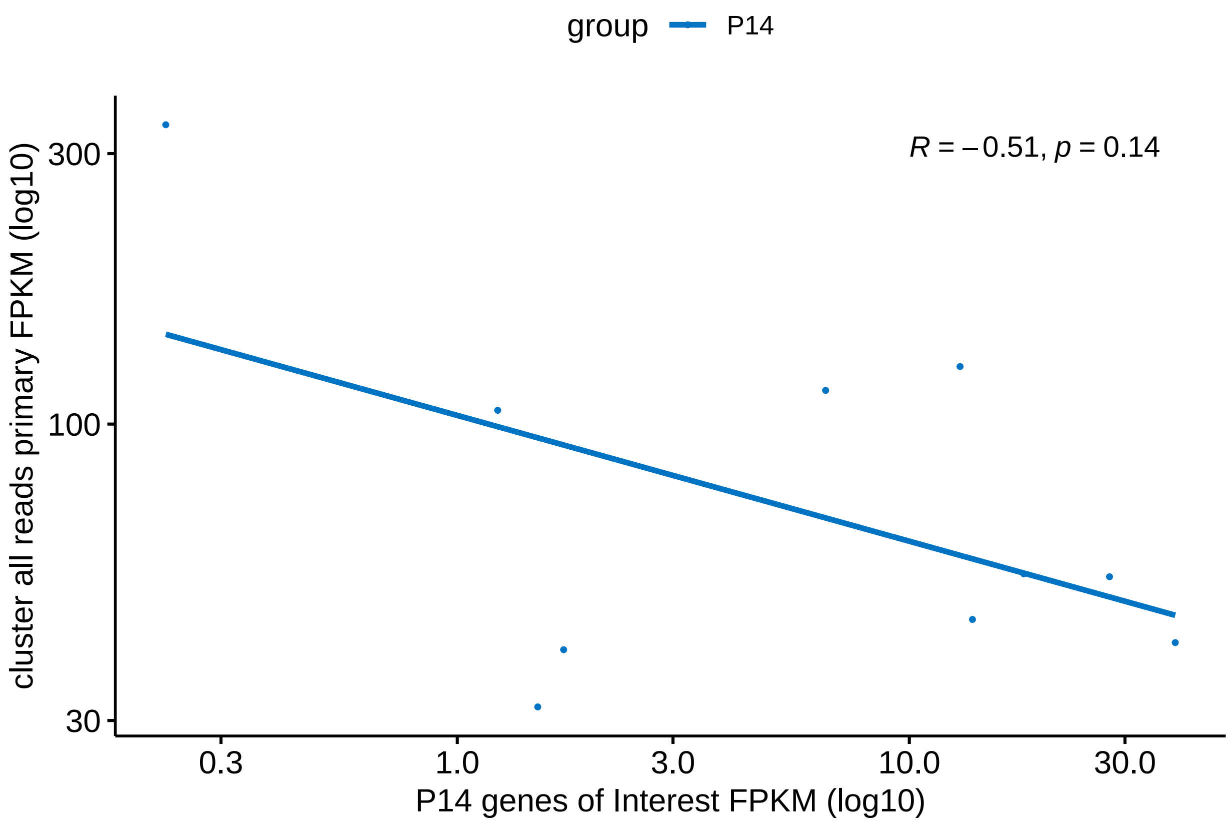
GoI = Top 10 genes

Make cluster expression tibble

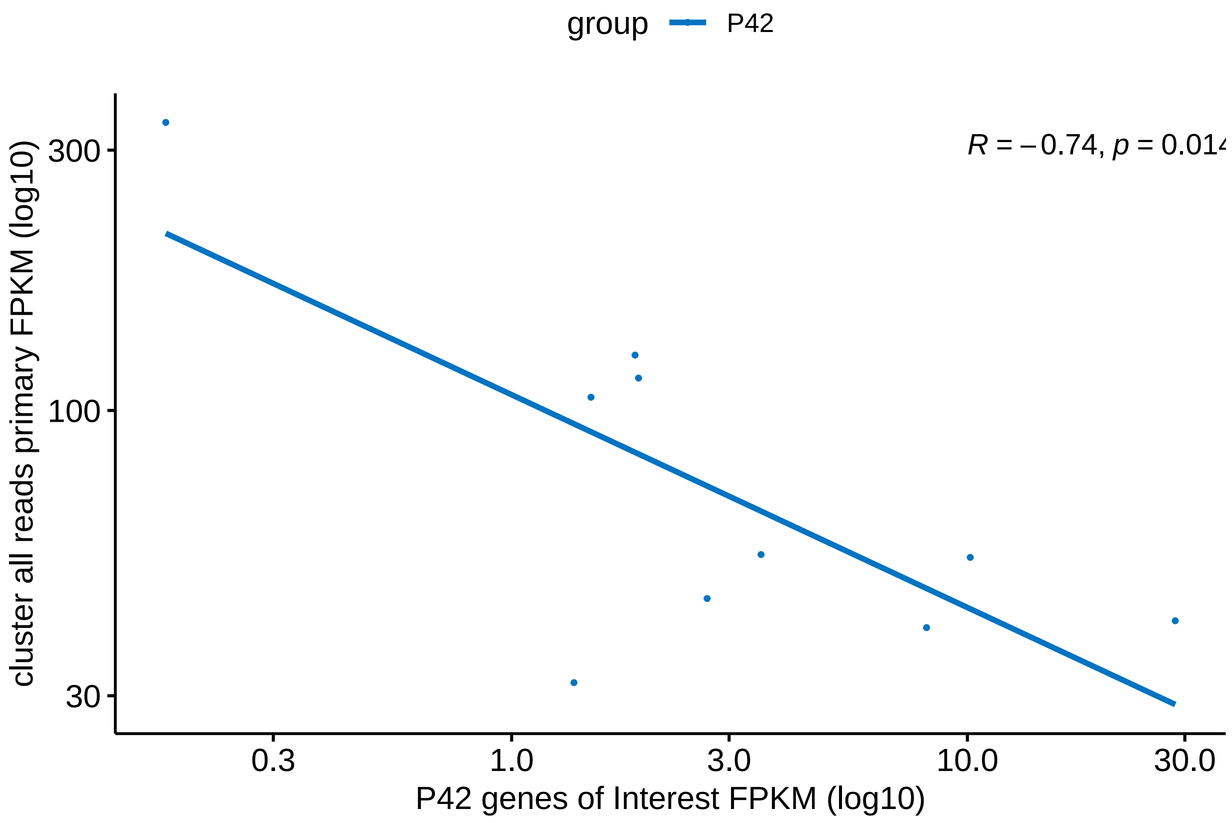
```
M <- Mili_prepach_gene3end_regions
clust_expr_fpkm.tb <-  tibble(gene_id=sapply(M$Gene_ovrlp,"[[",1), fpkm=M$all_reads_primary_alignments_F

# Collapse clusters mapping to the same gene and add up theirs fpkms
clust_expr_fpkm.tb <- clust_expr_fpkm.tb %>% group_by(gene_id) %>% summarise(fpkm=sum(fpkm))

E16.counts.GoI_clust <- inner_join(x = E16.counts.GoI, y = clust_expr_fpkm.tb, by = "gene_id") %>% muta

P14.counts.GoI_clust <- inner_join(x = P14.counts.GoI, y = clust_expr_fpkm.tb, by = "gene_id") %>% muta

P42.counts.GoI_clust <- inner_join(x = P42.counts.GoI, y = clust_expr_fpkm.tb, by = "gene_id") %>% muta

e16.p <- E16.counts.GoI_clust %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "E16 genes of Interest FPKM (log10)",
                                ylab = "cluster all reads primary FPKM (log10)",
                                size = 1,
                                shape = 20, palette = "jco",
                                add = "reg.line", color = "group"
                                ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

p14.p <- P14.counts.GoI_clust %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "P14 genes of Interest FPKM (log10)",
                                ylab = "cluster all reads primary FPKM (log10)",
                                size = 1,
                                shape = 20, palette = "jco",
                                add = "reg.line", color = "group"
                                ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

p42.p <- P42.counts.GoI_clust %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "P42 genes of Interest FPKM (log10)",
                                ylab = "cluster all reads primary FPKM (log10)",
                                size = 1,
                                shape = 20, palette = "jco",
                                add = "reg.line", color = "group"
                                ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

ggsave2(filename = "E16Int_vs_clusters.pdf", plot = e16.p, path = "./Plots")
ggsave2(filename = "P14Int_vs_clusters.pdf", plot = p14.p, path = "./Plots")
ggsave2(filename = "P42Int_vs_clusters.pdf", plot = p42.p, path = "./Plots")

e16.p
```



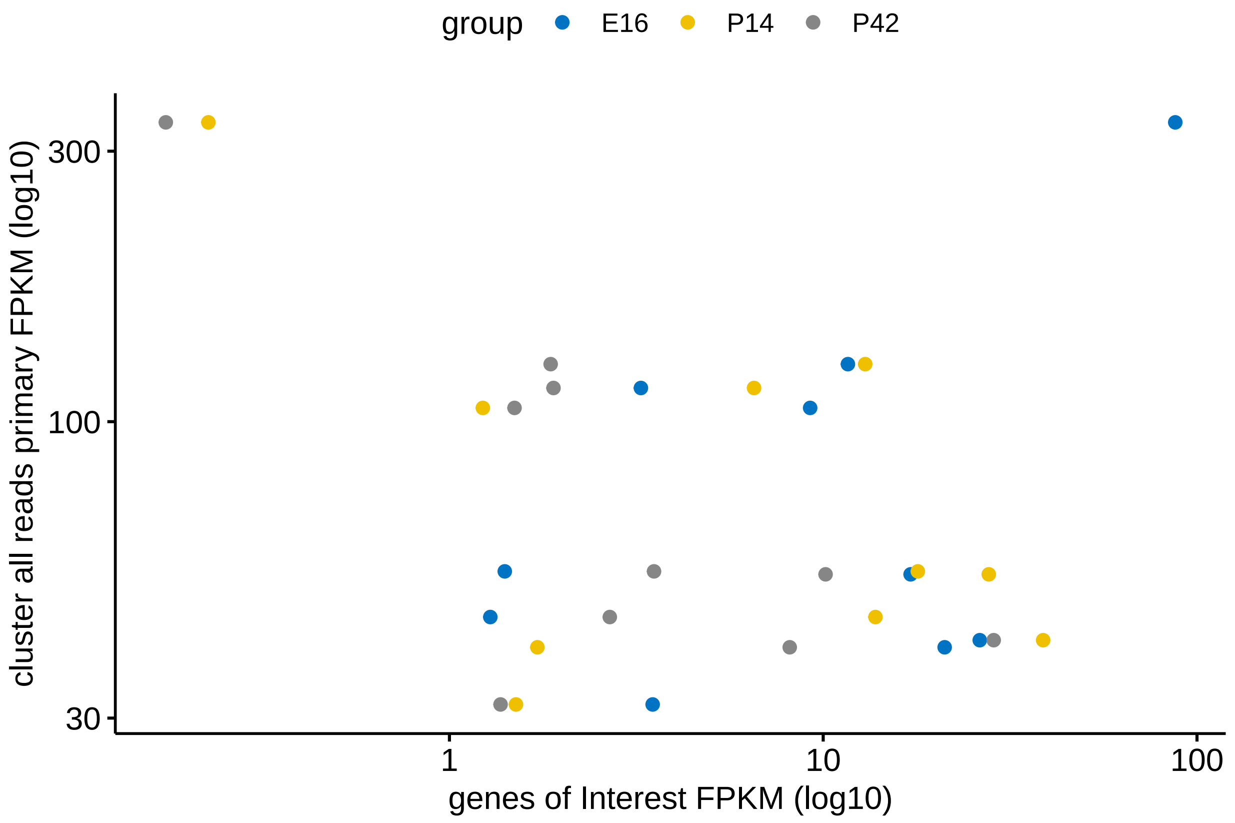
```
p14.p
```


group
P14
R = −0.51, p = 0.14
300
100
30
0.3
1.0
3.0
10.0
30.0
P14 genes of Interest FPKM (log10)
cluster all reads primary FPKM (log10)

```
p42.p
```


group
P42
R = −0.74, p = 0.01
300
100
30
0.3
1.0
3.0
10.0
30.0
P42 genes of Interest FPKM (log10)
cluster all reads primary FPKM (log10)

```
E16_P14_P42.counts.GoI_clust <- rbind(E16.counts.GoI_clust, P14.counts.GoI_clust, P42.counts.GoI_clust)

e16p14p42.p <- E16_P14_P42.counts.GoI_clust %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "genes of Interest FPKM (log10)",
                                ylab = "cluster all reads primary FPKM (log10)",
                                size = 3,
                                shape = 20, palette = "jco",
                                color = "group"
                                ) + scale_x_log10() + scale_y_log10()

ggsave2(filename = "E16P14P42Int_vs_clusters.pdf", plot = e16p14p42.p, path = "./Plots")
e16p14p42.p
```



Quantify RNAseq-based expression at cluster's intervals

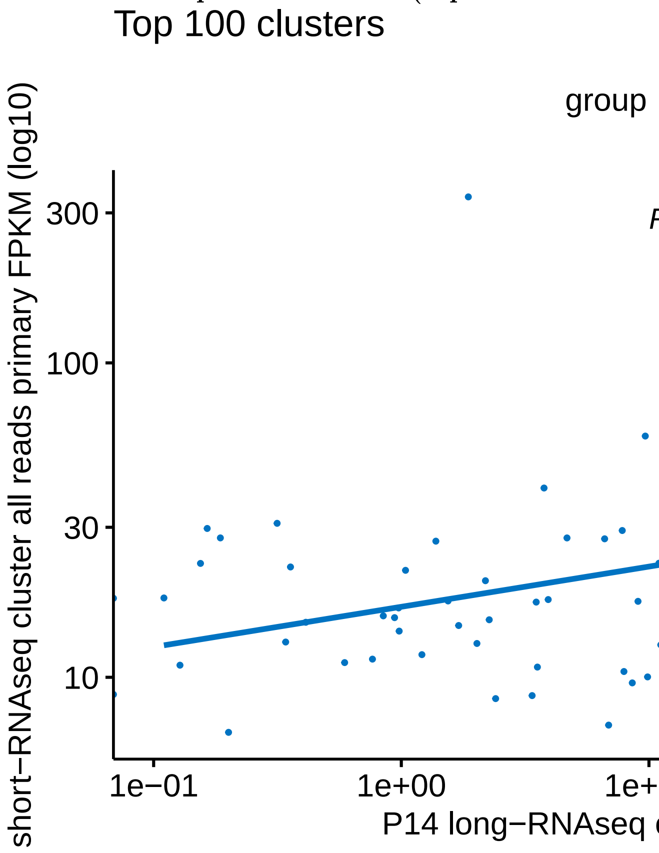
```
P14.cluster.norm.counts <- normalize_by_TPM(read_counts_column = P14.cluster.counts$counts,
                               annotation_file = ann.df,
                               annot_file_format = 'custom')

P42.cluster.norm.counts <- normalize_by_TPM(read_counts_column = P42.cluster.counts$counts,
                               annotation_file = ann.df,
                               annot_file_format = 'custom')

E16.cluster.norm.counts <- normalize_by_TPM(read_counts_column = E16.cluster.counts$counts,
                               annotation_file = ann.df,
                                annot_file_format = 'custom')
# Merge cluster expression tibble with long-RNAseq-cluster norm counts
P14.counts.long_short_clust <- inner_join(x = P14.cluster.norm.counts, y = clust_expr_fpkm.tb, by = "ge

P42.counts.long_short_clust <- inner_join(x = P42.cluster.norm.counts, y = clust_expr_fpkm.tb, by = "ge

E16.counts.long_short_clust <- inner_join(x = E16.cluster.norm.counts, y = clust_expr_fpkm.tb, by = "ge
```

Normalize cluster's read counts

```

# Plot correlations
p14.clust_vs_clust.p1 <- head(P14.counts.long_short_clust, n=100) %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "P14 long-RNAseq cluster FPKM (log10)",
                                ylab = "short-RNAseq cluster all reads primary FPKM (log10)",
                                title = "Top 100 clusters",
                                size = 1,
                                shape = 20, palette = "jco",
                                add = "reg.line", color = "group"
                                ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

p42.clust_vs_clust.p1 <- head(P42.counts.long_short_clust, n=100) %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "P42 long-RNAseq cluster FPKM (log10)",
                                ylab = "short-RNAseq cluster all reads primary FPKM (log10)",
                                title = "Top 100 clusters",
                                size = 1,
                                shape = 20, palette = "jco",
                                add = "reg.line", color = "group"
                                ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

e16.clust_vs_clust.p1 <- head(E16.counts.long_short_clust, n=100) %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "E16 long-RNAseq cluster FPKM (log10)",
                                ylab = "short-RNAseq cluster all reads primary FPKM (log10)",
                                title = "Top 100 clusters",
                                size = 1,
                                shape = 20, palette = "jco",
                                add = "reg.line", color = "group"
                                ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

# Append E16, P42 and P14 tibbles
E16_P14_P42.counts.long_short_clust <- rbind(head(E16.counts.long_short_clust, n=100), head(P14.counts.l

e16p14p42.clust_vs_clust.p1 <- E16_P14_P42.counts.long_short_clust %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "long-RNAseq cluster FPKM (log10)",
                                ylab = "short-RNAseq cluster all reads primary FPKM (log10)",
                                title = "Top 100 clusters",
                                size = 3,
                                shape = 20, palette = "jco",
                                color = "group"
```

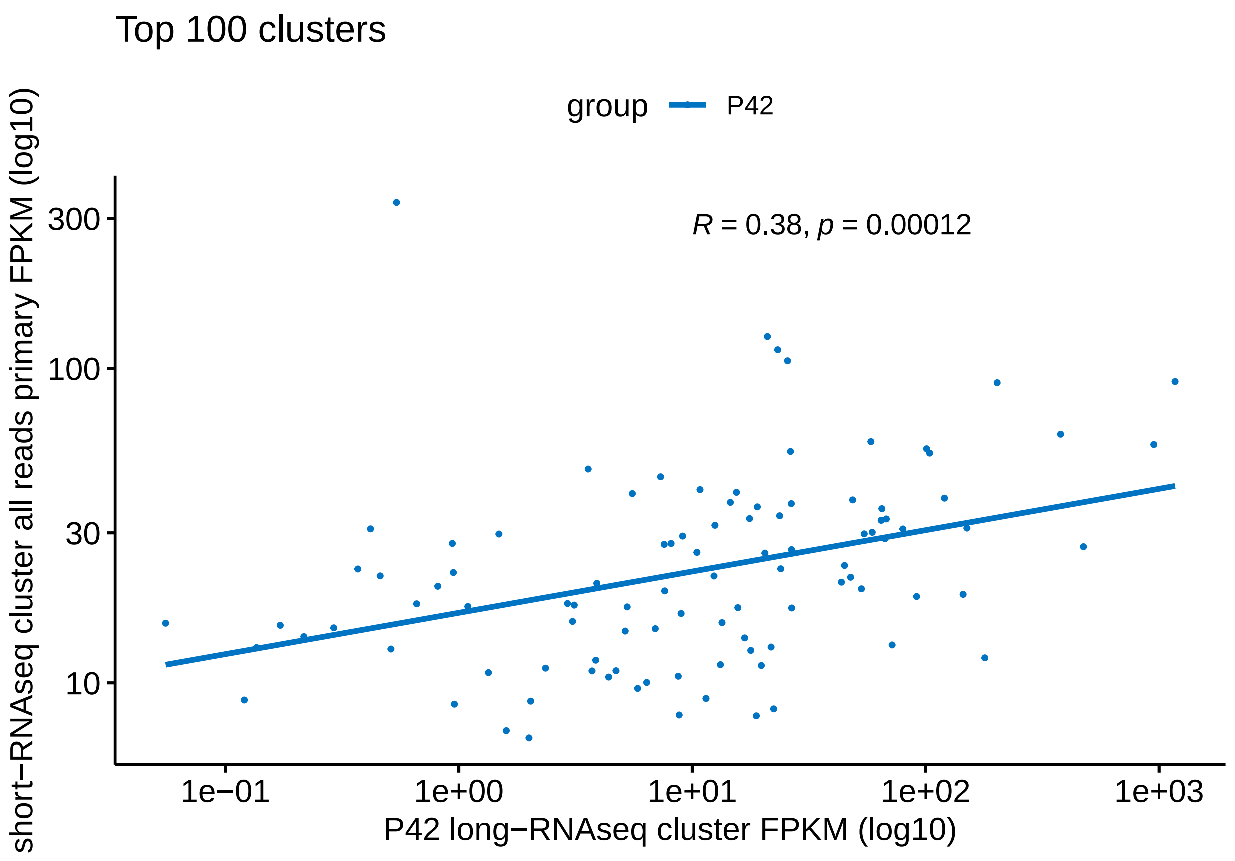
```
                                    ) + scale_x_log10() + scale_y_log10()

ggsave2(filename = "P14_long_vs_short_clust.pdf", plot = p14.clust_vs_clust.p1, path = "./Plots")
ggsave2(filename = "P42_long_vs_short_clust.pdf", plot = p42.clust_vs_clust.p1, path = "./Plots")
ggsave2(filename = "E16_long_vs_short_clust.pdf", plot = e16.clust_vs_clust.p1, path = "./Plots")
ggsave2(filename = "E16P14P42_long_vs_short_clust.pdf", plot = e16p14p42.clust_vs_clust.p1, path = "./Pl

p14.clust_vs_clust.p1
```

Plot correlations between long-RNAseq-based clusters and short-RNAseq-based clusters (top

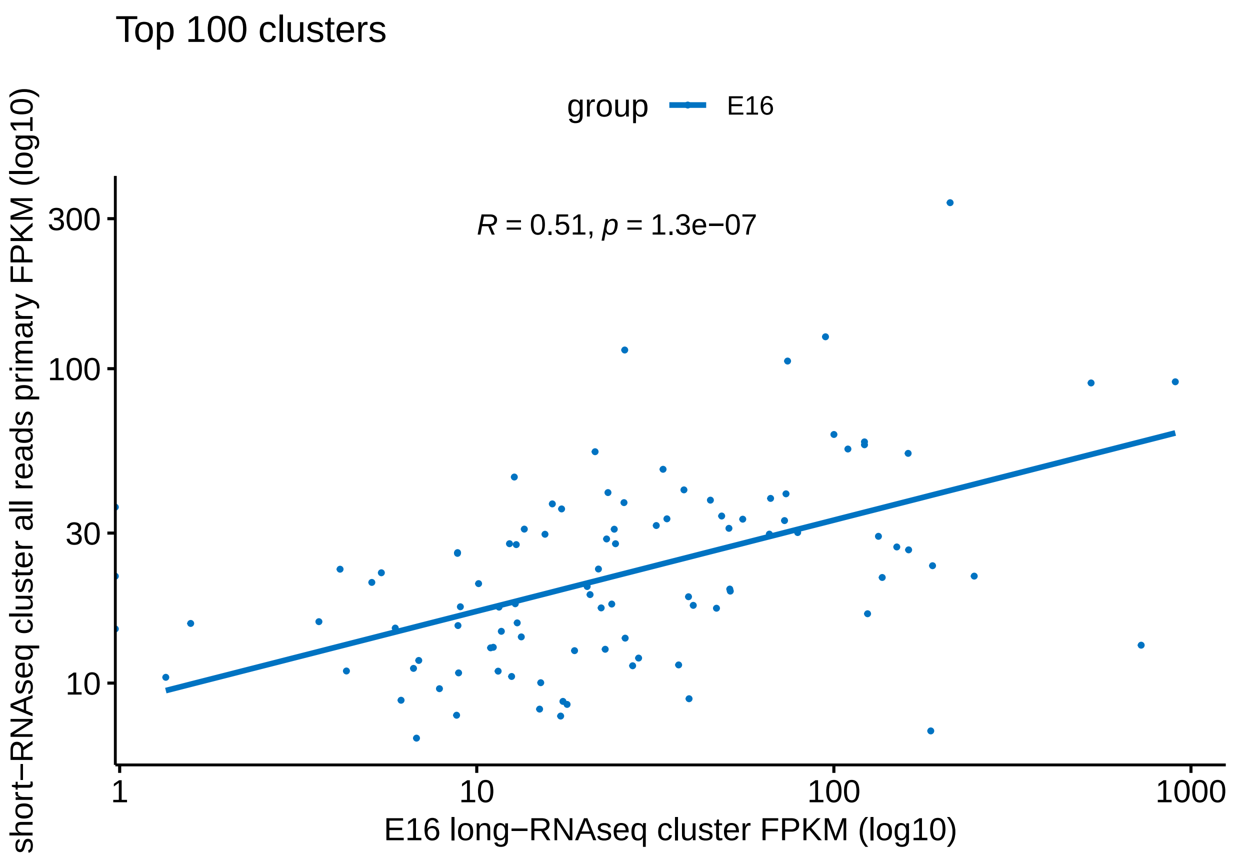


100 clusters sorted based on their FPM counts for piRNAs)

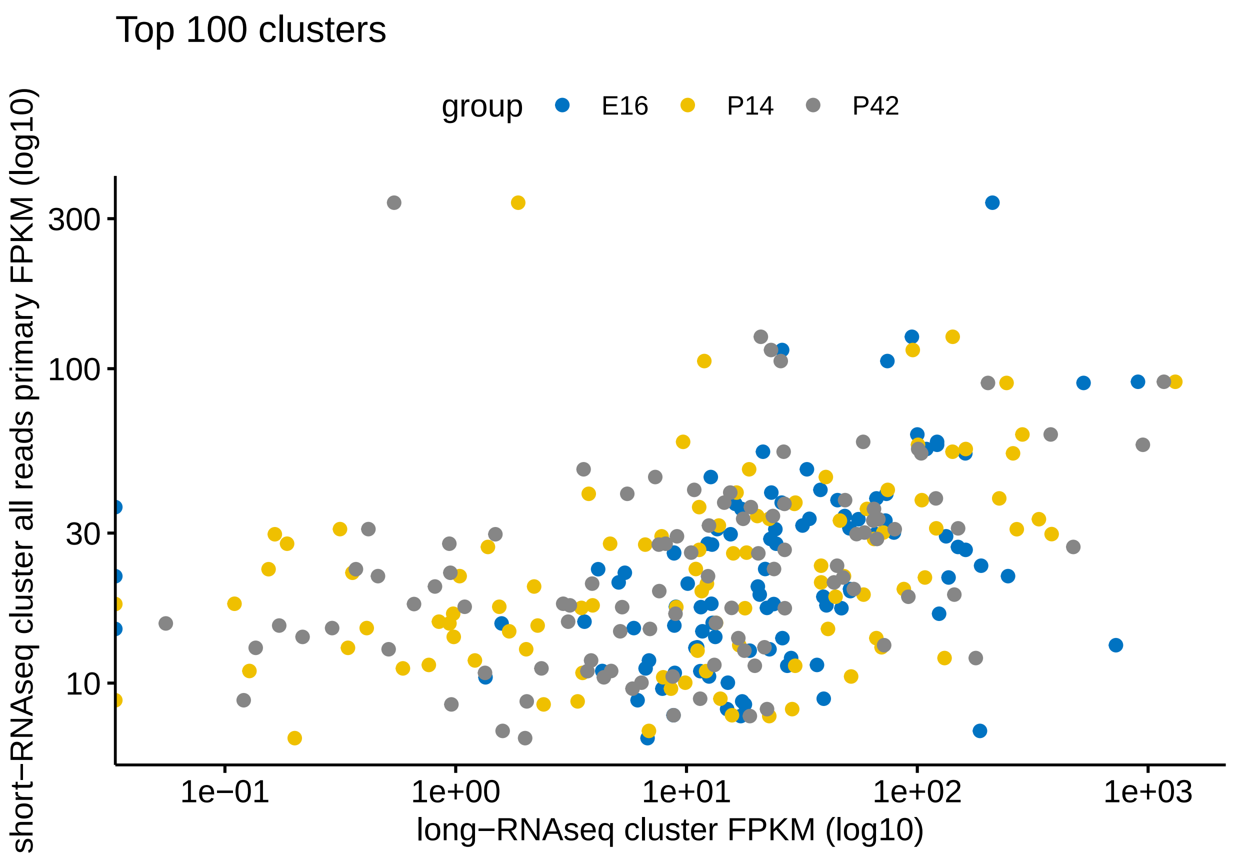
```
p42.clust_vs_clust.p1
```



```
e16.clust_vs_clust.p1
```



```
e16p14p42.clust_vs_clust.p1
```

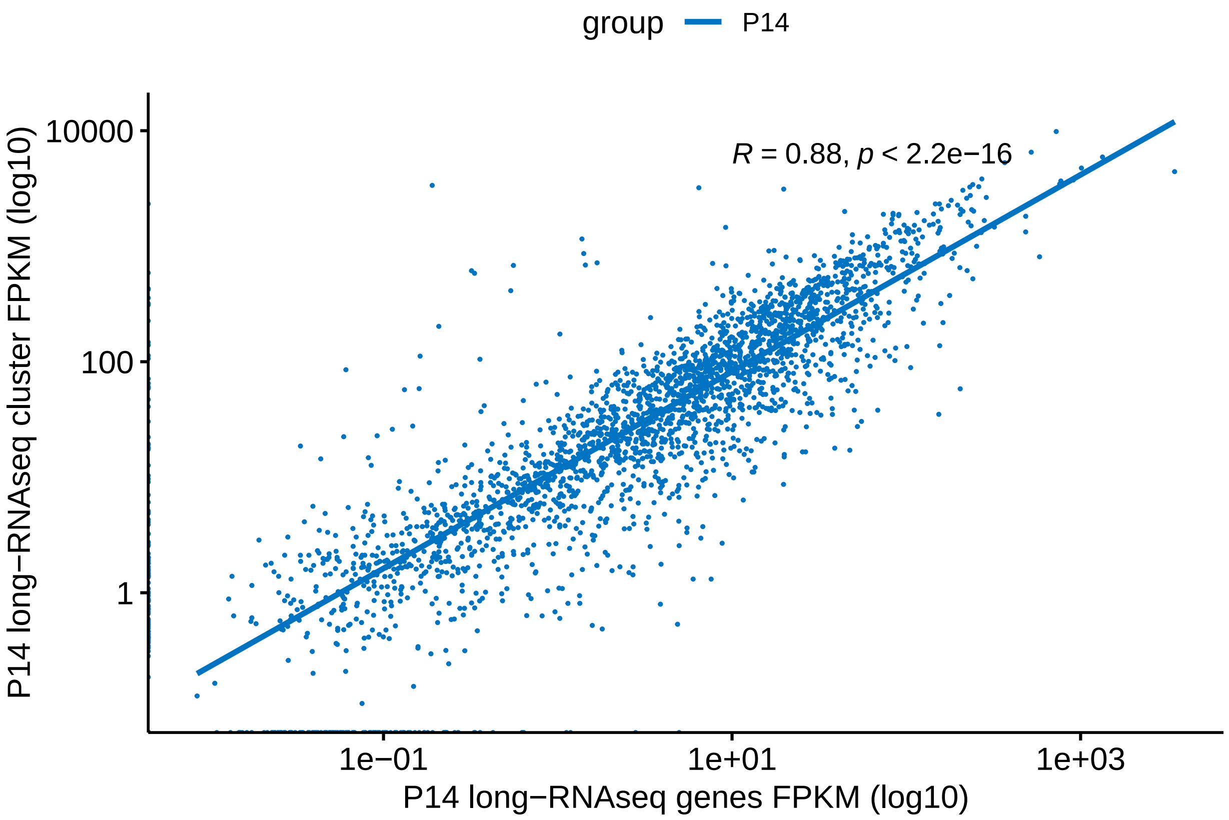


```
p14_long_genes_clusters <- inner_join(x = P14.counts, y = P14.cluster.norm.counts, by = "gene_id") %>% r

p42_long_genes_clusters <- inner_join(x = P42.counts, y = P42.cluster.norm.counts, by = "gene_id") %>% r

e16_long_genes_clusters <- inner_join(x = E16.counts, y = E16.cluster.norm.counts, by = "gene_id") %>% r
```

Correlation between long-RNAseq genes and long-RNAseq clusters

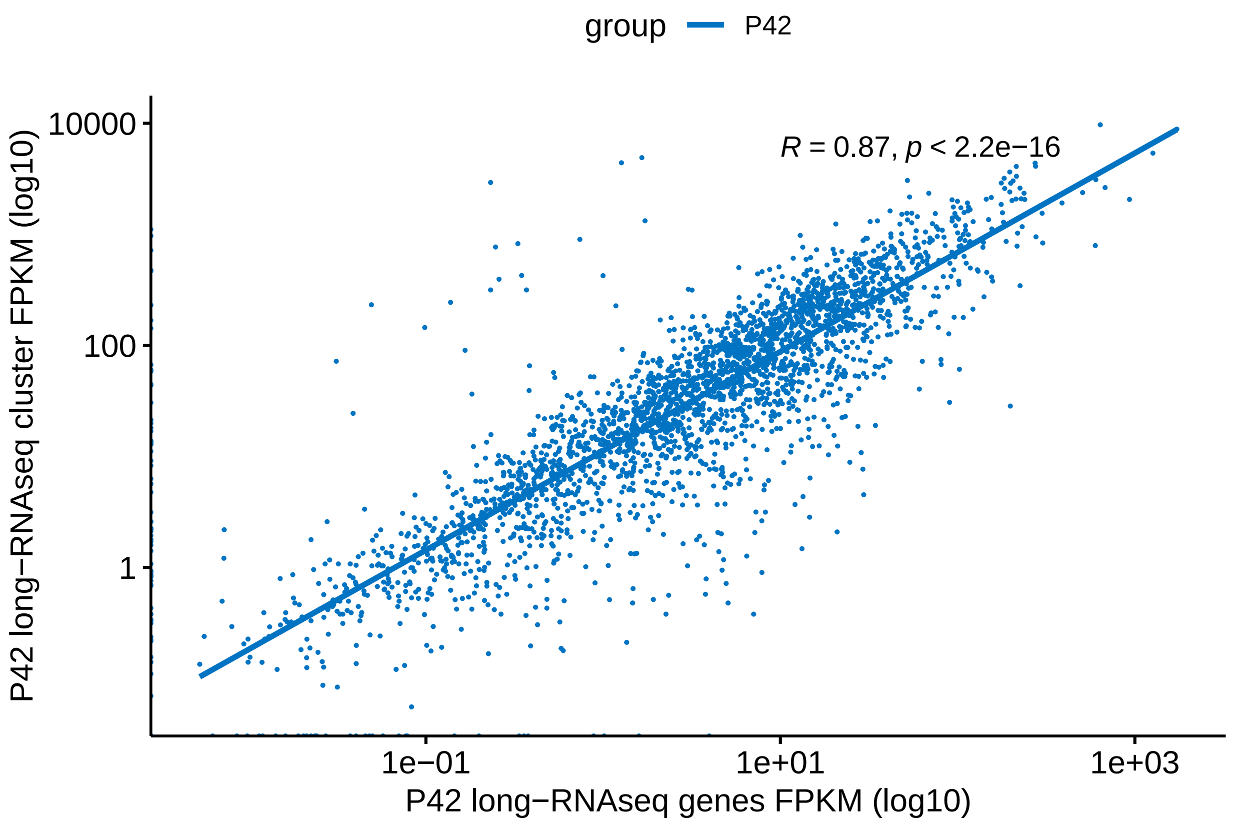
```
p14.long_genes_clusters.p1 <- p14_long_genes_clusters %>% ggscatter(x = "fpkm.x",
                                  y = "fpkm.y",
                                  xlab = "P14 long-RNAseq genes FPKM (log10)",
                                  ylab = "P14 long-RNAseq cluster FPKM (log10)",
                                  size = 0.5,
                                  shape = 20, palette = "jco",
                                  add = "reg.line", color = "group"
                                  ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

p24.long_genes_clusters.p1 <- p42_long_genes_clusters %>% ggscatter(x = "fpkm.x",
                                  y = "fpkm.y",
                                  xlab = "P42 long-RNAseq genes FPKM (log10)",
                                  ylab = "P42 long-RNAseq cluster FPKM (log10)",
                                  size = 0.5,
                                  shape = 20, palette = "jco",
                                  add = "reg.line", color = "group"
```

```
                                ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

e16.long_genes_clusters.p1 <- e16_long_genes_clusters %>% ggscatter(x = "fpkm.x",
                                y = "fpkm.y",
                                xlab = "E16 long-RNAseq genes FPKM (log10)",
                                ylab = "E16 long-RNAseq cluster FPKM (log10)",
                                size = 0.5,
                                shape = 20, palette = "jco",
                                add = "reg.line", color = "group"
                                ) + scale_x_log10() + scale_y_log10() + stat_cor(label.x = 1)

ggsave2(filename = "P14_long_genes_clusters.pdf", plot = p14.long_genes_clusters.p1, path = "./Plots", 

ggsave2(filename = "P42_long_genes_clusters.pdf", plot = p24.long_genes_clusters.p1, path = "./Plots", 

ggsave2(filename = "E16_long_genes_clusters.pdf", plot = e16.long_genes_clusters.p1, path = "./Plots")

p14.long_genes_clusters.p1
```

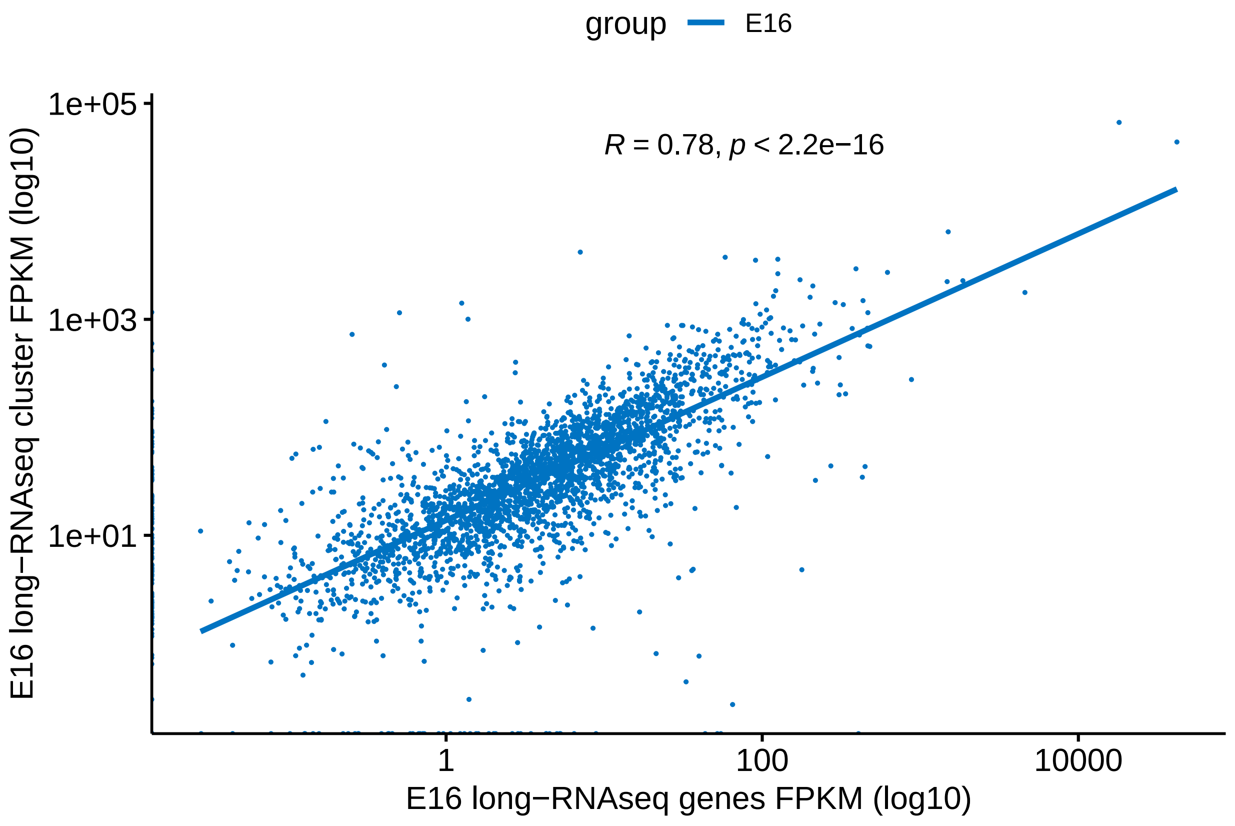


Plots

```
p24.long_genes_clusters.p1
```



```
e16.long_genes_clusters.p1
```



Questions derived from meeting with Astrid

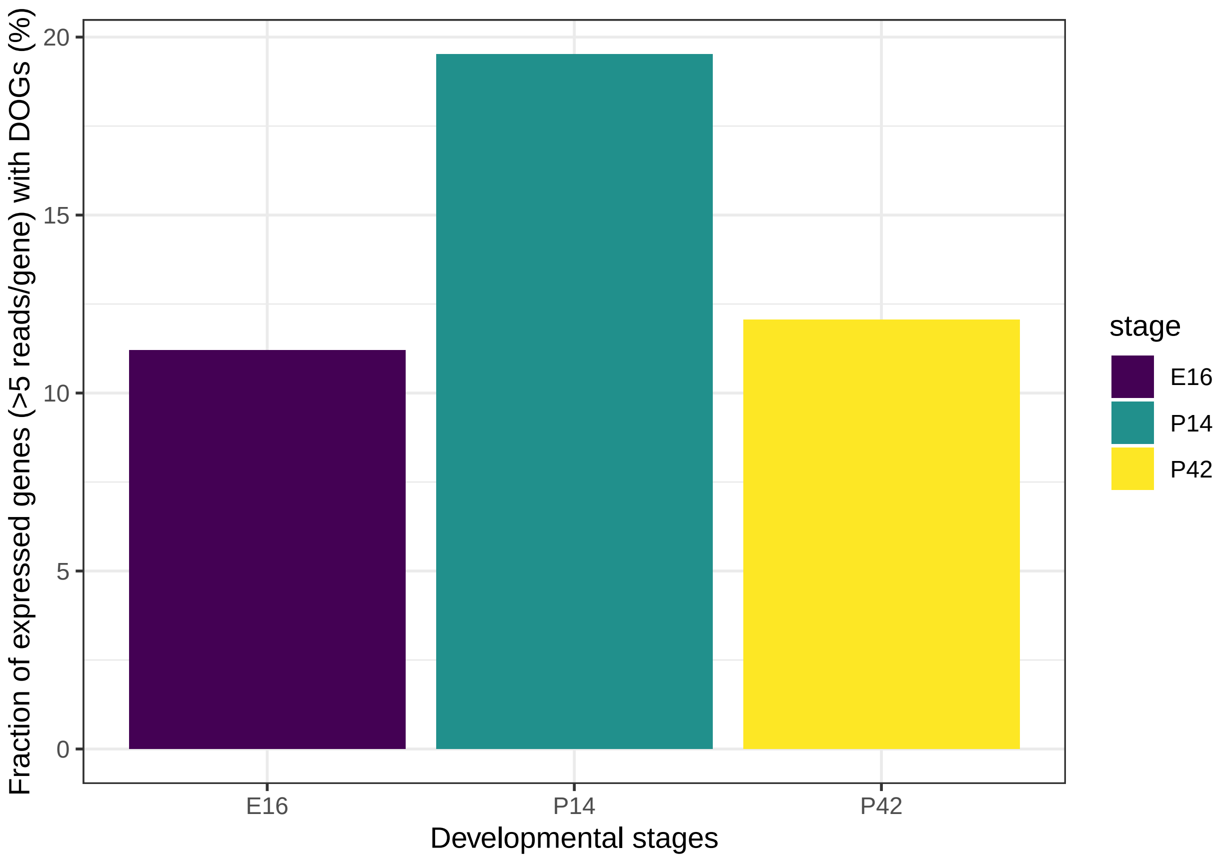
```
# Load predicted DOGs
dog_prediction_path <- "/Users/lorenziha/Documents/DKBIOCORE_LOCAL/TK_59/NEW_ARTDECO_ANALYSIS/ARTDECO_DI

P14.dogs <- read_delim(file = paste0(dog_prediction_path, "P14.dogs.fpkm.txt"), col_names = c("gene_id"

P42.dogs <- read_delim(file = paste0(dog_prediction_path, "P42.dogs.fpkm.txt"), col_names = c("gene_id"

E16.dogs <- read_delim(file = paste0(dog_prediction_path, "E16_5.dogs.fpkm.txt"), col_names = c("gene_i
```

```
P14.expressed_genes <- filter(P14.counts, read_counts > 5)
P42.expressed_genes <- filter(P42.counts, read_counts > 5)
E16.expressed_genes <- filter(E16.counts, read_counts > 5)

results <- tibble(Description="Fraction of expressed genes (> 5 reads per gene) with predicted DOGs (%)
       P14=round(100*length(P14.dogs$gene_id)/length(P14.expressed_genes$gene_id), 2),
       P42=round(100*length(P42.dogs$gene_id)/length(P42.expressed_genes$gene_id), 2),
       E16=round(100*length(E16.dogs$gene_id)/length(E16.expressed_genes$gene_id), 2)
       )

# results %>% ggplot(aes(bam, value, fill=bam)) + geom_bar(stat="identity") + facet_wrap(~bam_file, dir

results.tbl <-  tibble(stage=colnames(results)[2:4],values=t(results[,2:4])[,1])

p <- results.tbl %>% ggplot(aes(stage, values, fill=stage)) + geom_bar(stat="identity") + theme_bw() + 

ggsave2(filename = "./Plots/fract_exp_genes_with_dogs.pdf", plot = p, width = 4, height = 8)

print(results)
```

1- Fraction of expressed genes (> 5 reads per gene) with DOGs for each developmental stage.

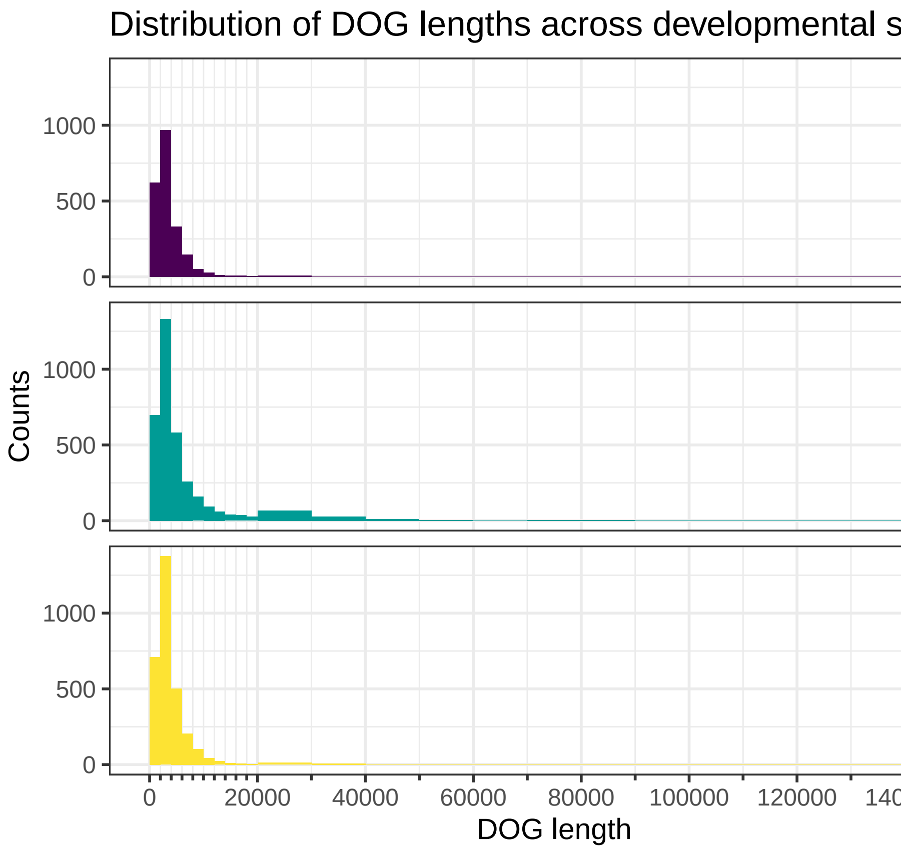
```
## # A tibble: 1 x 4
##   Description                                               P14   P42   E16
##   <chr>                                                   <dbl> <dbl> <dbl>
## 1 Fraction of expressed genes (> 5 reads per gene) with predi~  19.5  12.1  11.2
```

```
p
```



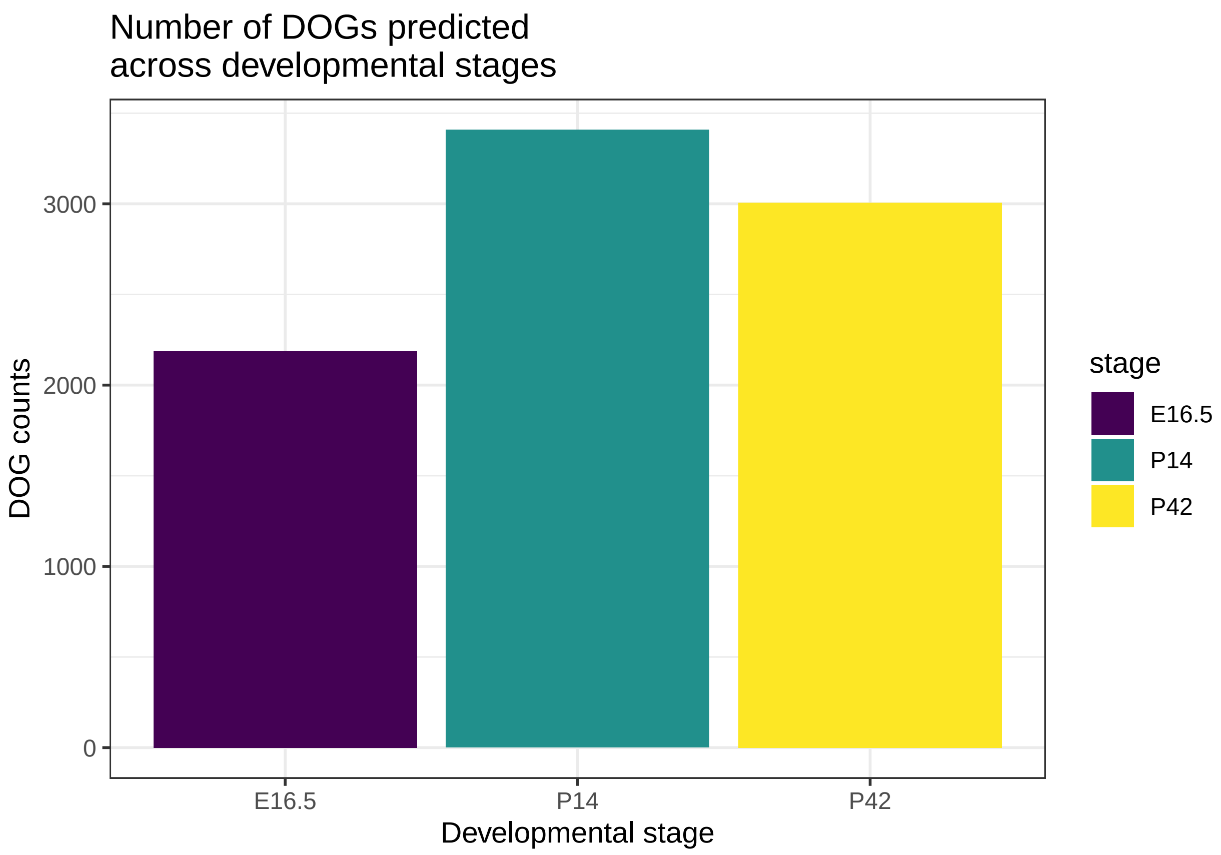
```
library(easyGgplot2)
library(ggprism)
require(graphics)

# Append dog predictions for all developmental stages
all_stages.dogs <- tibble(
    rbind(mutate(P14.dogs, stage="P14"),
          mutate(P42.dogs, stage="P42"),
          mutate(E16.dogs, stage="E16.5")
        )
    )

hist.p1 <- ggplot(all_stages.dogs,aes(x=length, fill=stage)) +
  geom_histogram(breaks=c(seq(0,19999,2000),seq(20000,150000,10000))) +
  scale_x_continuous(breaks=seq(0,150000,20000),
                     minor_breaks = c(seq(0,19999,2000),seq(20000,150000,10000)),
                     guide = "prism_minor"
                     ) +
  xlab("DOG length") + ylab("Counts") + labs(title = "Distribution of DOG lengths across developmental s
  facet_wrap(~stage, dir="v", strip.position="right") +
  theme_bw() + scale_fill_manual(values=hcl.colors(3, "viridis"))


ggsave2(filename = "./Plots/dog_distrib_1.pdf", plot = hist.p1)
```

```
hist.p1
```

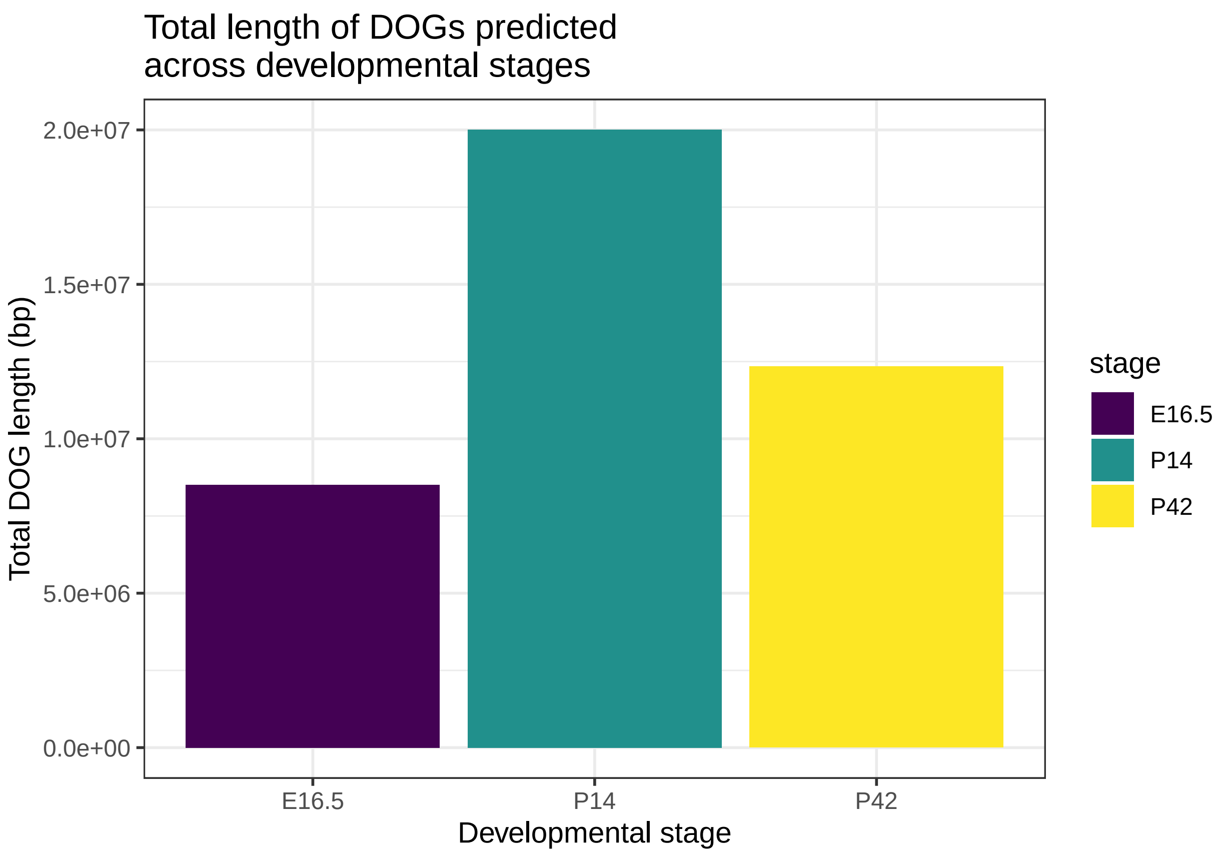


2- Distribution of DOGs across their size
#### Summary of predicted DOGs per developmental stage

```
dog_summary_counts.p <- all_stages.dogs %>% ggplot(aes(x=stage, fill=stage)) +
    geom_bar() +
    xlab("Developmental stage") + ylab("DOG counts") +
    labs(title = "Number of DOGs predicted\nacross developmental stages") +
    theme_bw() + scale_fill_viridis_d(option = "D")

dog_summary_sums.p <-all_stages.dogs %>% ggplot(aes(x=stage, y=length, fill=stage, color=NULL)) +
    geom_bar(stat="identity") +
    xlab("Developmental stage") + ylab("Total DOG length (bp)") +
    labs(title = "Total length of DOGs predicted\nacross developmental stages") +
    theme_bw() + scale_fill_viridis_d(option = "D")

ggsave2(filename = "./Plots/dog_counts_per_stage.pdf", plot = dog_summary_counts.p, width = 4, height =
ggsave2(filename = "./Plots/total_dog_lengths_per_stage.pdf", plot = dog_summary_sums.p, width = 4, heig

dog_summary_counts.p
```

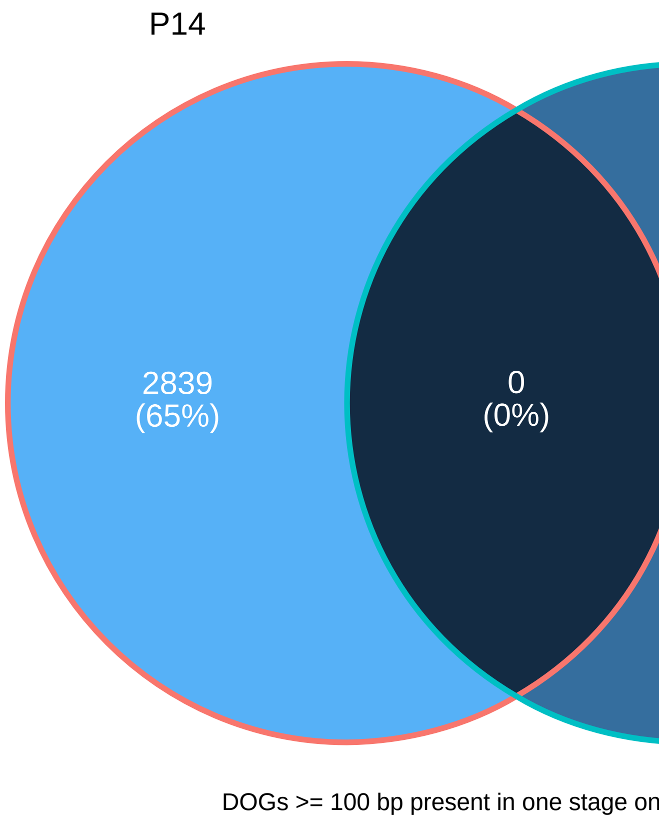


```
dog_summary_sums.p
```



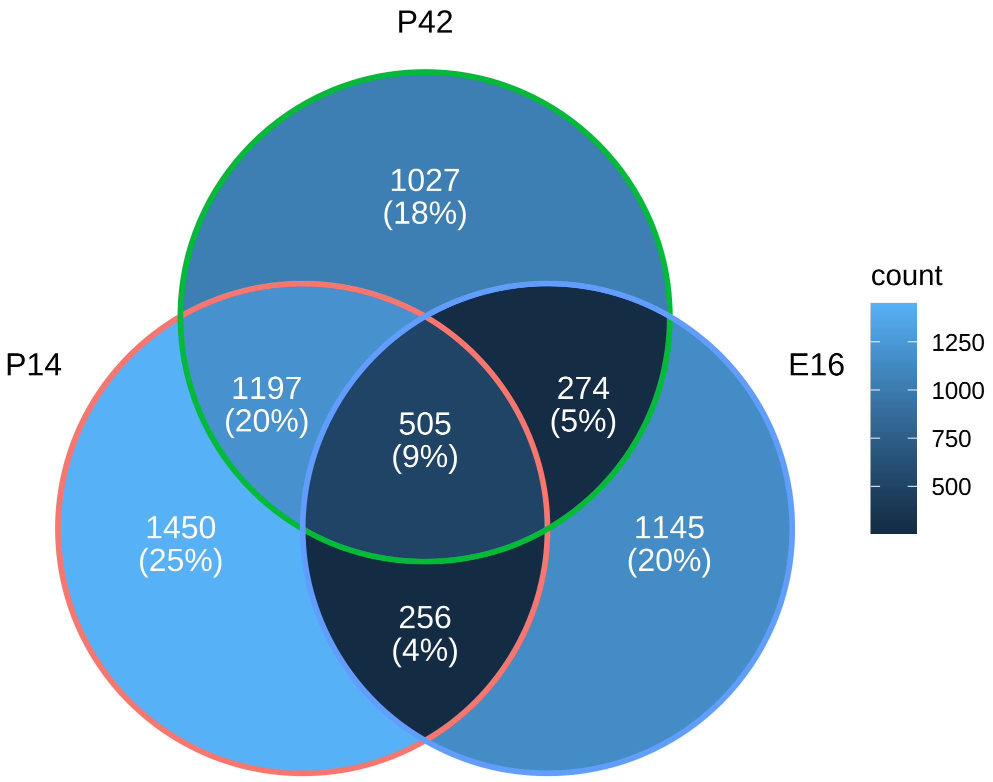
```
library("ggVennDiagram")

all_stages.across.dogs <- full_join(P14.dogs, P42.dogs, by="gene_id") %>% full_join(E16.dogs, by="gene_
colnames(all_stages.across.dogs) <- c("gene_id","length.P14","fpkm.P14","length.P42","fpkm.P42","length

all_stages.across.dogs <- all_stages.across.dogs %>%
                              mutate(P14_E16=log2(length.P14/length.E16)) %>%
                              mutate(P42_E16=log2(length.P42/length.E16)) %>%
                              mutate(P14_P42=log2(length.P14/length.P42))

venn <- list()
min_dog_length = 100
venn$P14 <- filter(all_stages.across.dogs, P14_E16 >= 1  | (length.P14 > min_dog_length & is.na(length.
venn$E16 <- filter(all_stages.across.dogs, P14_E16 <= -1 | (length.E16 > min_dog_length & is.na(length.

venn_1.p <- ggVennDiagram(venn,label_alpha = 0, label_color = "white") +
  #ggplot2::scale_fill_gradient(low="purple3",high = "yellow3") +
  labs(caption = "DOGs >= 100 bp present in one stage only or twice as long as in the other stage")


venn_all.1000 <- list()
min_dog_length = 1000
venn_all.1000$P14 <- filter(all_stages.across.dogs, length.P14 > min_dog_length )$gene_id
venn_all.1000$P42 <- filter(all_stages.across.dogs, length.P42 > min_dog_length )$gene_id
venn_all.1000$E16 <- filter(all_stages.across.dogs, length.E16 > min_dog_length )$gene_id
```

```r
venn_all.p <- ggVennDiagram(venn_all.1000,label_alpha = 0, label_color = "white") +
  #ggplot2::scale_fill_gradient(low="purple3",high = "yellow3") +
  labs(caption = "DOGs > 1000 bp")

ggsave2(filename = "./Plots/comparative_venn_p14_e16.pdf", plot = venn_1.p)
ggsave2(filename = "./Plots/comparative_venn_p14_e16_p42_dog1000.pdf", plot = venn_all.p)

venn_1.p
```



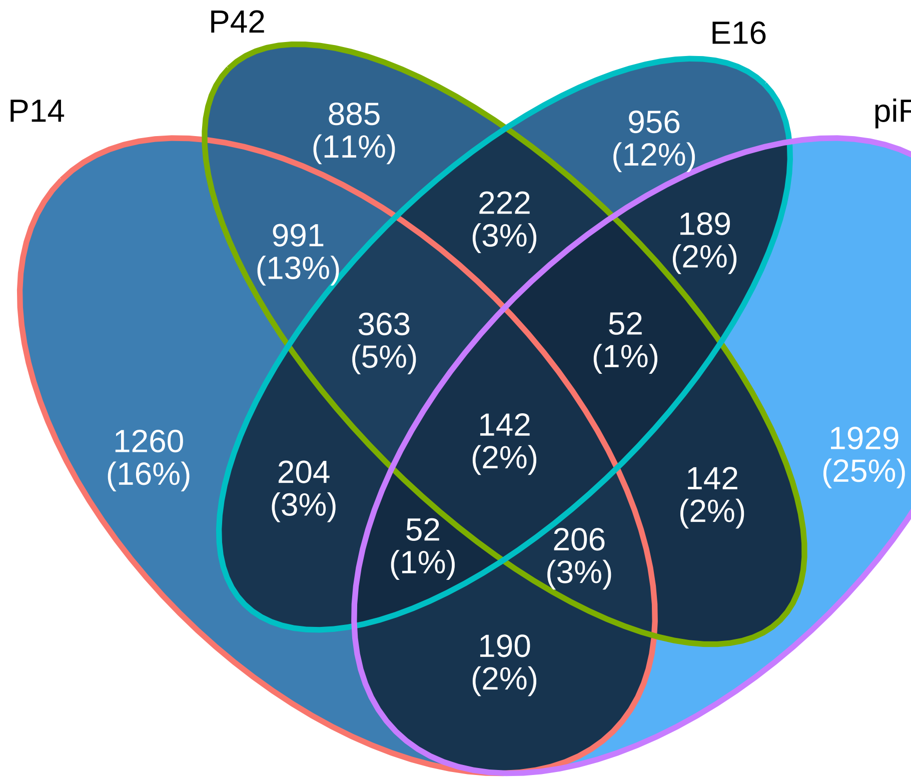
Venn diagram of shared DOGs across developmental stages

```r
venn_all.p
```



DOGs > 1000 bp

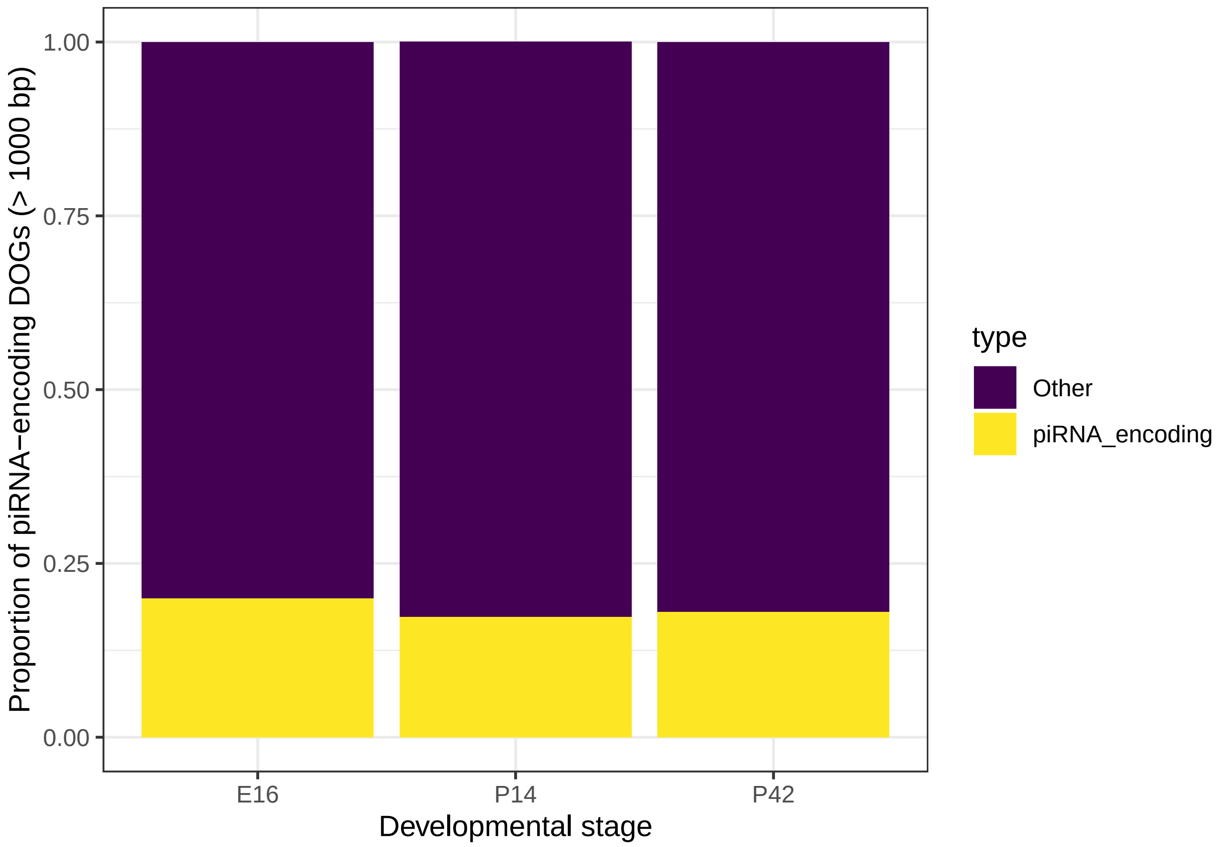
```
# As venn diagram
venn_fraction <- venn_all.1000
venn_fraction$piRNA_clusters <- sapply(M$Gene_ovrlp,"[[",1)

venn_fraction.p <- ggVennDiagram(venn_fraction,label_alpha = 0, label_color = "white") +
  #scale_fill_viridis_b(option = "D") + # ggplot2::scale_fill_gradient(low="purple3",high = "yellow3")
  labs(caption = "DOGs > 1000 bp")

ggsave2(filename = "./Plots/comparative_venn_p14_e16_p42_fraction_piRNA_clusters.pdf",
        plot = venn_fraction.p)

venn_fraction.p
```



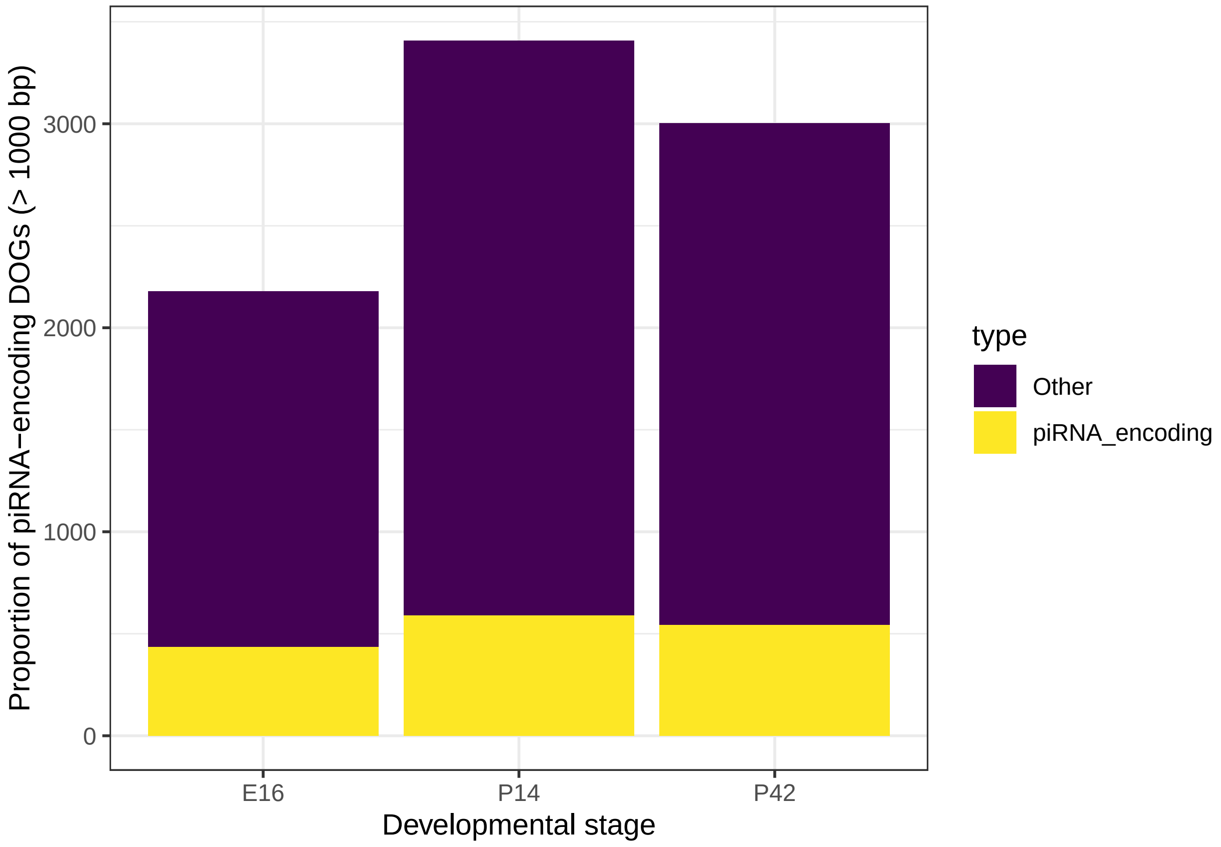
Fraction of DOGs that produce piRNAs

```
# As bar plot

x <- rbind(
  tibble(type=ifelse(venn_fraction$P14 %in% venn_fraction$piRNA_clusters,"piRNA_encoding","Other"), stag
  tibble(type=ifelse(venn_fraction$P42 %in% venn_fraction$piRNA_clusters,"piRNA_encoding","Other"), stag
  tibble(type=ifelse(venn_fraction$E16 %in% venn_fraction$piRNA_clusters,"piRNA_encoding","Other"), stag
)

barchar_proportion.p <- x %>% ggplot(aes(stage, fill=type)) + geom_bar(position = "fill") +
  scale_fill_viridis_d(option = "D") +
  ylab("Proportion of piRNA-encoding DOGs (> 1000 bp)") +
  xlab("Developmental stage") +
  theme_bw()

barchar_count.p <- x %>% ggplot(aes(stage, fill=type)) + geom_bar() +
  scale_fill_viridis_d(option = "D") +
  ylab("Proportion of piRNA-encoding DOGs (> 1000 bp)") +
  xlab("Developmental stage") +
  theme_bw()


ggsave2(filename = "./Plots/comparative_barchar_p14_e16_p42_proportion_of_piRNA_clusters.pdf",
        plot = barchar_proportion.p, width = 4, height = 8)

ggsave2(filename = "./Plots/comparative_barchar_p14_e16_p42_count_of_piRNA_clusters.pdf",
        plot = barchar_count.p, width = 4, height = 8)
```

```
barchar_proportion.p
```



```
barchar_count.p
```



```
my_predictions_dir <- "NEW_ARTDECO_ANALYSIS/ARTDECO_DIR_FPKM_0.003_dog2kb_wind500bp_dogcov0.05_doglen2k
e16.tbl <- read_delim(file = paste0("./",my_predictions_dir,"/dogs/E16_5.dogs.bed"), col_names = c("Chr
p14.tbl <- read_delim(file = paste0("./",my_predictions_dir,"/dogs/P14.dogs.bed"), col_names = c("Chrom
p42.tbl <- read_delim(file = paste0("./",my_predictions_dir,"/dogs/P42.dogs.bed"), col_names = c("Chrom


e16.tbl <- e16.tbl %>% mutate(Dev_stage = "E16")
p14.tbl <- p14.tbl %>% mutate(Dev_stage = "P14")
p42.tbl <- p42.tbl %>% mutate(Dev_stage = "P42")

all.tbl <- e16.tbl %>% bind_rows(p14.tbl, p42.tbl)
all.tbl <- all.tbl %>% mutate(Length=End3-End5)
```

Quantify total number of predicted DOGs per developmental stage

```
p4 <- all.tbl %>% group_by(Dev_stage, Chrom) %>% count() %>%
  ggbarplot(x="Dev_stage", y="n", ylab = "Number of predicted DOGs", xlab = "Developmental stage",
            fill = "Dev_stage", ggtheme = theme_classic(), facet.by = "Chrom") +
    scale_fill_viridis_d(option = "D")

ggsave2(filename = "./Plots/Fig_4.pdf", plot = p4, width = 5, height = 8)
p4
```

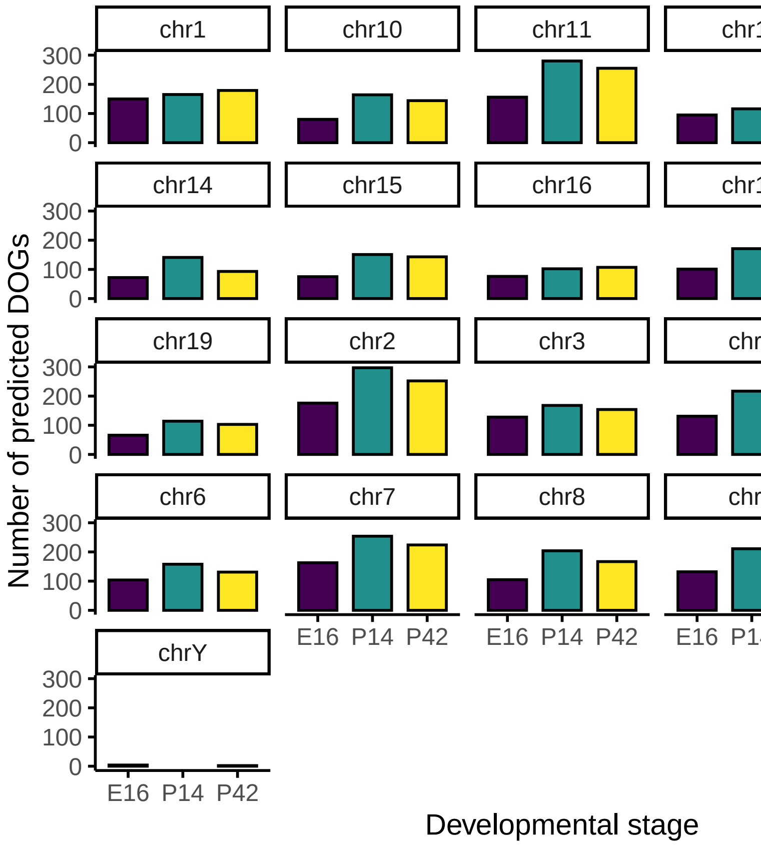


Fig 4: Number of predicted DOGs per chromosome

```
total_counts <- list()
e16_total_dog_counts <- filter(all.tbl %>% group_by(Dev_stage) %>% count(), Dev_stage=="E16")$n
p14_total_dog_counts <- filter(all.tbl %>% group_by(Dev_stage) %>% count(), Dev_stage=="P14")$n
p42_total_dog_counts <- filter(all.tbl %>% group_by(Dev_stage) %>% count(), Dev_stage=="P42")$n
total_counts['E16'] <- e16_total_dog_counts
total_counts['P14'] <- p14_total_dog_counts
total_counts['P42'] <- p42_total_dog_counts

p5 <- all.tbl %>% group_by(Dev_stage, Chrom) %>% count() %>% mutate(Total_counts = total_counts[[Dev_sta
  ggbarplot(x="Dev_stage", y="Norm_counts", ylab = "Normalized number of predicted DOGs", xlab = "Devel
            fill = "Dev_stage", ggtheme = theme_classic(), facet.by = "Chrom") +
    scale_fill_viridis_d(option = "D")

ggsave2(filename = "./Plots/Fig_5.pdf", plot = p5, width = 5, height = 8)
p5
```

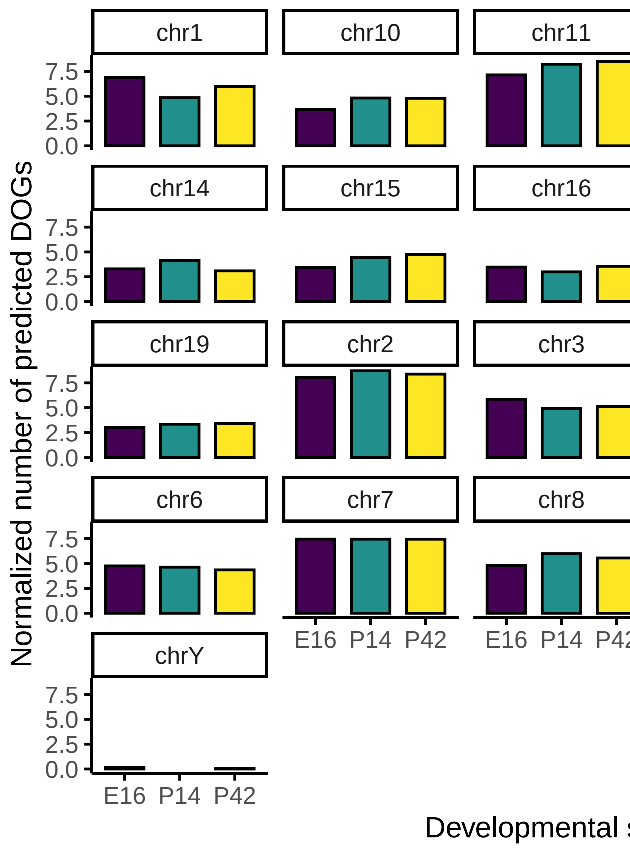


Fig 5: Normalized counts of predicted DOGs per chromosome

```r
save.image(file = "./data/tk_59_environment.Rdata")
```

Save project's data

```r
sessionInfo()
```

R session information

```
## R version 4.3.1 (2023-06-16)
## Platform: x86_64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;  LAPACK
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices datasets  utils     methods
## [8] base
```

```
##
## other attached packages:
##  [1] parseR_0.1.0           edgeR_3.42.4           limma_3.56.2
##  [4] scales_1.2.1           ggVennDiagram_1.2.3    ggprism_1.0.4
##  [7] easyGgplot2_1.0.0.9000 ggpubr_0.6.0           Rsubread_2.14.2
## [10] GenomicFeatures_1.52.1 AnnotationDbi_1.62.2   Biobase_2.60.0
## [13] DGEobj.utils_1.0.6     lubridate_1.9.2        forcats_1.0.0
## [16] stringr_1.5.0          dplyr_1.1.2            purrr_1.0.2
## [19] readr_2.1.4            tidyr_1.3.0            tibble_3.2.1
## [22] ggplot2_3.4.3          tidyverse_2.0.0        cowplot_1.1.1
## [25] GenomicRanges_1.52.0   GenomeInfoDb_1.36.4    IRanges_2.34.1
## [28] S4Vectors_0.38.2       BiocGenerics_0.46.0
##
## loaded via a namespace (and not attached):
##   [1] RColorBrewer_1.1-3          shape_1.4.6
##   [3] rstudioapi_0.15.0           magrittr_2.0.3
##   [5] farver_2.1.1                rmarkdown_2.24
##   [7] ragg_1.2.5                  GlobalOptions_0.1.2
##   [9] BiocIO_1.10.0               zlibbioc_1.46.0
##  [11] vctrs_0.6.3                 memoise_2.0.1
##  [13] Rsamtools_2.16.0            RCurl_1.98-1.12
##  [15] rstatix_0.7.2               htmltools_0.5.6
##  [17] S4Arrays_1.0.5              progress_1.2.2
##  [19] curl_5.0.2                  broom_1.0.5
##  [21] KernSmooth_2.23-21          plyr_1.8.9
##  [23] cachem_1.0.8                GenomicAlignments_1.36.0
##  [25] DGEobj_1.1.2                lifecycle_1.0.3
##  [27] iterators_1.0.14            pkgconfig_2.0.3
##  [29] Matrix_1.6-1                R6_2.5.1
##  [31] fastmap_1.1.1               clue_0.3-65
##  [33] GenomeInfoDbData_1.2.10     MatrixGenerics_1.12.3
##  [35] digest_0.6.33               colorspace_2.1-0
##  [37] DESeq2_1.40.2               textshaping_0.3.6
##  [39] RSQLite_2.3.1               labeling_0.4.3
##  [41] filelock_1.0.2              fansi_1.0.4
##  [43] timechange_0.2.0            mgcv_1.9-0
##  [45] httr_1.4.7                  abind_1.4-5
##  [47] compiler_4.3.1              proxy_0.4-27
##  [49] bit64_4.0.5                 withr_2.5.0
##  [51] doParallel_1.0.17           backports_1.4.1
##  [53] BiocParallel_1.34.2         carData_3.0-5
##  [55] DBI_1.1.3                   highr_0.10
##  [57] ggsignif_0.6.4              biomaRt_2.56.1
##  [59] rappdirs_0.3.3              DelayedArray_0.26.7
##  [61] classInt_0.4-10             rjson_0.2.21
##  [63] ggsci_3.0.0                 units_0.8-4
##  [65] tools_4.3.1                 glue_1.6.2
##  [67] restfulr_0.0.15             nlme_3.1-163
##  [69] sf_1.0-14                   grid_4.3.1
##  [71] reshape2_1.4.4              cluster_2.1.4
##  [73] generics_0.1.3              gtable_0.3.4
##  [75] tzdb_0.4.0                  class_7.3-22
##  [77] data.table_1.14.8           hms_1.1.3
##  [79] xml2_1.3.5                  car_3.1-2
##  [81] utf8_1.2.3                 XVector_0.40.0
##  [83] foreach_1.5.2              pillar_1.9.0
##  [85] vroom_1.6.3                splines_4.3.1
##  [87] circlize_0.4.15            BiocFileCache_2.8.0
##  [89] lattice_0.21-8             renv_1.0.2
##  [91] rtracklayer_1.60.1         bit_4.0.5
##  [93] tidyselect_1.2.0           ComplexHeatmap_2.16.0
##  [95] locfit_1.5-9.8             Biostrings_2.68.1
##  [97] knitr_1.43                 SummarizedExperiment_1.30.2
##  [99] xfun_0.40                  matrixStats_1.0.0
## [101] stringi_1.7.12             yaml_2.3.7
## [103] evaluate_0.21              codetools_0.2-19
## [105] BiocManager_1.30.22        RVenn_1.1.0
## [107] cli_3.6.1                  systemfonts_1.0.4
## [109] munsell_0.5.0              Rcpp_1.0.11
## [111] dbplyr_2.3.3               png_0.1-8
## [113] XML_3.99-0.14              parallel_4.3.1
## [115] assertthat_0.2.1           blob_1.2.4
## [117] prettyunits_1.1.1          bitops_1.0-7
## [119] viridisLite_0.4.2          ggthemes_4.2.4
## [121] e1071_1.7-13               crayon_1.5.2
## [123] GetoptLong_1.0.5           rlang_1.1.1
## [125] KEGGREST_1.40.0
```